Crosscut Resilience and Power (BOG 13)

ASCR Workshop on Extreme Heterogeneity in HPC
Jan 23-25, 2018

Connect here:

Zoom: https://lbnl.zoom.us/j/8903175440
Telephone: US: +1 646 558 8656 or +1 669 900 6833
Meeting ID: 890 317 5440

(These) slides here:

https://docs.google.com/presentation/d/16H9R-F5vyTNhnNH1SNUAW1lYv_fWRMacXKXdXJXKkE0/edit?usp=sharing

Crosscut Resilience and Power Contributors (30-35)

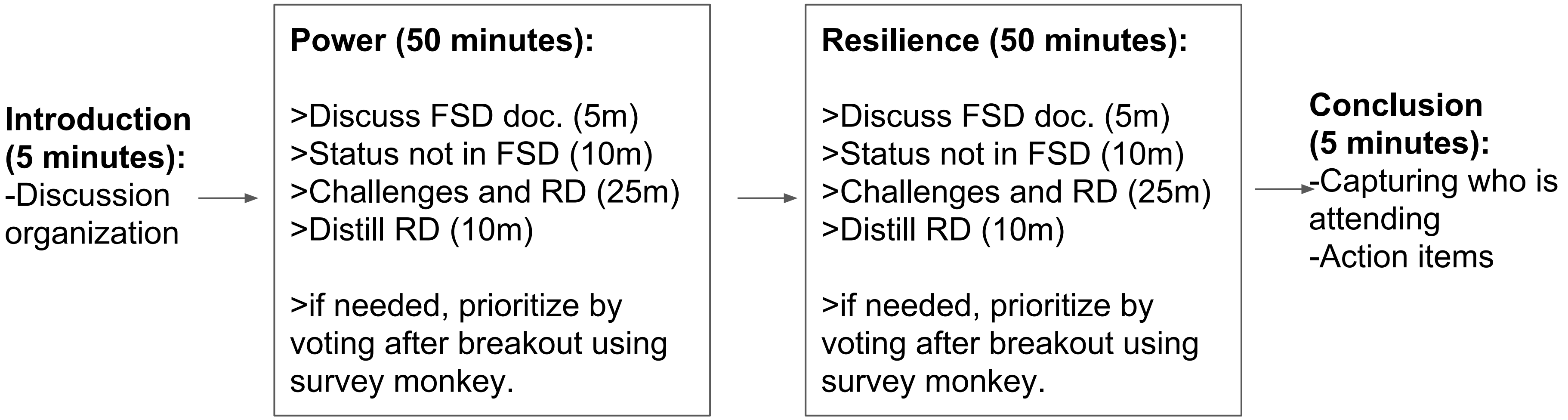
Moderator(s): Franck Cappello (ANL), Kirk Cameron (Virginia Tech)

Full list

- Mattan Erez
- Anshu Dubey
- Sriram Krishnamoorthy
- Kevin Barker
- Andrs Marquez
- Leon Song
- Candy Culhane
- Anastasiia Butko
- Christian Engelmann
- Dilip Vasudevan
- Esam El-Araby
- Jeanine Cook
- Katie Schuman
- Keita Teranishi
- Kevin Barker
- Kevin Pedretti
- Larry Kaplan
- Paul Peltz
- Peter Kogge
- Ron Brightwell
- Barry Rountree
- Shirley Moore
- Tiffany Mintz
- Zhiling Lan
- Pete Beckman
- Kamil Iskra
- Vitus Leung
- Andreas Gerstlauer
- Antonino Tumeo

Plan

Goal for this breakout: Identify N possible/promising research directions that address key challenges for DOE mission in the 2030+ timeframe. Focus on aspects related to heterogeneity.



Introduction (5 minutes):
-Discussion organization

Power (50 minutes):

>Discuss FSD doc. (5m)
>Status not in FSD (10m)
>Challenges and RD (25m)
>Distill RD (10m)

>if needed, prioritize by voting after breakout using survey monkey.

Lead: Kirk
Secretary: Leon
Timer: Franck

Resilience (50 minutes):

>Discuss FSD doc. (5m)
>Status not in FSD (10m)
>Challenges and RD (25m)
>Distill RD (10m)

>if needed, prioritize by voting after breakout using survey monkey.

Lead: Franck
Secretary: Christian
Timer: Kirk

Conclusion (5 minutes):
-Capturing who is attending
-Action items

Crosscut Resilience and Power Contributors

Moderator(s): Franck Cappello (ANL), Kirk Cameron (Virginia Tech)

BOGists:

- Christian Engelmann (ORNL), scribe for Resilience
- Leon Song (PNNL), scribe for Power
- Larry Kaplan (Cray)
- Sriram Krishnamoorthy (PNNL)
- Zhiling Lan (Illinois Tech)
- Barry Rountree (LLNL)
- Vitus Leung (SNL)
- Esam El-Araby (KU)

Reminder of Our Charge

The purpose of this workshop is to identify the priority research directions for ASCR in providing a smart software stack that includes techniques, such as deep learning to make future computers composed of a variety of complex processors, new interconnects and deep memory hierarchies easily used by a broad community of computational scientists.

Focus on the challenges of Extreme Heterogeneity: why does topic X get better/worse due to Extremely Heterogeneous architectures

Power Capability Targets for Extreme Heterogeneity

CHARGE: Stretch our thinking beyond exascale (2020's) to post-Moore (2030's)
(12-->20 years from now)

The purpose of the workshop is to define the challenges that extreme heterogeneity presents to the software stack including the programming environment, and to identify priority research directions in computer science that are essential to making **extremely heterogeneous systems usable**, **efficient**, and secure for science applications and DOE mission requirements.

BOG 13 EH Power Targets for 2030

Target 1: Maximize top supercomputer performance under a 20 MW ceiling.

Target 2: Maximize system power efficiency to enable highest performance.

2017 #1 Top500: 93 petaflops, 15 MW (6,051 MFlops/Watt)
2027 Projected: +/- 1 exaflops, 20 MW (50,000 MFlops/Watt)
2037 Projected: +/- 10 exaflops, 20->30 MW (333k->500k MFlops/Watt)

**projections based on Top500/Green500 trends from 2007-->2017.

BOG 13 Current Status for EH Power (TRENDS)

Performance heterogeneity related to power (1.1.4 (pg 5))

- EE algorithm designs (sparse, adaptive, irregular, imprecise)
- EE fine-grain power management in HW (dark silicon, clock gating, low-threshold)
  - Includes emergent memory technologies (1.1.5, pg 6)
- EE coarse-grain power management in software (throttling)
  - Includes heterogeneity from non-volatile memories (1.1.5 (pg 7))
- Heterogeneous power management (including non-CMOS)
  - "Non-CMOS will augment not replace CMOS." --Bob Colwell.

(Maybe several viewgraphs)

BOG 13 Current Status for EH Power (CONSEQUENCES)

Extreme performance variability and heterogeneity.

Further exposure of underlying heterogeneity to user.

New tradeoffs in latency, power, persistence, resilience.

Implications for hardware, systems software, libraries, applications.

Move from systems that draw some consistent percentage of TDP to systems with much wider power variation (although probably nowhere close to TDP). (rountree)

Lack of sufficient granularity monitoring and control. Lack of tools to do fine-grained SW based power optimization (kaplan)

(Maybe several viewgraphs)

BOG 13 Current Capability for EH Power Efficiency

Capability 1: Existing techniques are localized, reactive, lack adoption.

Capability 2: Existing approaches cannot hide all heterogeneity from user.

Capability 3: Existing infrastructure has difficulty dealing with system-wide linpack-sized power swings. Swings of similar magnitude in EH systems may occur multiple times per minute. (rountree)

BOG 13: list of key research challenges

Challenge 13.1 Adaptive: Adapt to optimize for efficiency in extreme heterogeneous environs
Challenge 13.2 Transparent: Hide details from the user to the extent possible
Challenge 13.3 Effective: Maximize performance under power constraints
Challenge 13.4 Holistic: Cross-stack solution optimizing across the system stack
Challenge 13.5 Consistent: Dependable, deterministic performance to encourage adoption
Challenge 13.6 Portable: Techniques that work across systems
Challenge 13.7 Even in the presence of abundant power, firmware-based within-node power/energy/thermal (PET) systems remain application-oblivious. (rountree)
Challenge 13.8 Ultimately PET optimization is constrained by APIs and capabilities of firmware. Improving these is nontrivial, usually for non-technical reasons. (rountree)
Challenge 13.9 Modeling of power, energy, thermals of large-scale systems (analytical modeling or cycle-accurate simulation). (song)

BOG 13 (EH Power) Target: ~10 Exaflops in under 30 MW.

Execute efficiently under a PET constraint. This will require advanced PET measurement, optimization, and control (MOC) in extremely heterogeneous environments. MOC solutions must be transparent to the user and portable and interoperable across EH systems. Transparent, portable and interoperable EH MOC challenges are of unprecedented scale and complexity potentially requiring adaptive, unsupervised, intelligent machine learning.

BOG 13 Possible Research Directions Summary

PRD 13.1 Studies of where performance and efficiency are best accomplished in the stack
- Includes 2.1.3 (pg 11) Compiler support for power/energy

PRD 13.2 New execution models
- Includes non-BSP models of execution
- Includes models of computation for new and non-CMOS technologies

PRD 13.3 Runtime systems for portable, power-performance efficiency
- Includes 2.1.4 (pg 13) Autotuning
- Includes 2.2 (pg 14) OS Management for power/energy
- Includes 2.2.1 (pg 16) Complex Resource Management
- Includes 2.2.1 (pg 16) Heterogeneous Memory Management
- Includes 2.2.1 (pg 16) Decentralized Resource Management for power/energy
- Includes 2.2.2 (pg 17) Support for dark silicon

PRD 13.4 New metrics for heterogeneous performance and power/energy
- Includes 2.4 (pg 19) new technologies and non-CMOS (neuromorphic, quantum)

PRD 13.5 New tools to quantify and evaluate performance, energy, and reliability
- Includes 2.8 (pg 28) Role of power/energy in extreme heterogeneity and specialization
- Includes 2.8.2 (pg 29) How to quantify power, energy, reliability

PRD 13.6 Synergistic co-design dialog with vendors (rountree)

PRD 13.7 Related Activity: 1.2 (pg 7) Related Activities. Beyond Moore Electronics

PRD 13.8 Cycle accurate simulation/modeling of power, energy, thermals.

BOG 13 (EH Power) Target: ~10 Exaflops in under 30 MW.

Execute efficiently under a PET constraint. This will require advanced PET measurement, optimization, and control (MOC) in extremely heterogeneous environments. MOC solutions must be transparent to the user and portable and interoperable across EH systems. Transparent, portable and interoperable EH MOC challenges are of unprecedented scale and complexity potentially requiring adaptive, unsupervised, intelligent machine learning.

PRD 13.1 EH PET runtime systems, compilers, resource management, and workflow automation including vendor co-design of hardware, software, and algorithms for MOC

PRD 13.2 EH PET simulation, modelling, measurement, and metrics for MOC

MOC = Measurement, Optimization, Control

PRD X.n : Short title of possible research direction

- One paragraph description (3 sentence/bullet)
- Research challenges
  - Metrics for progress
- Potential research approaches and research directions
- How and when will success impact technology?

[This slide is a place holder for now...once we converge, we can write these out.]

(Maybe several viewgraphs)

Resilience Status and Recent Advances in FSD

Resilience is mentioned in several places in the FSD. It is discussed as a **source of performance heterogeneity**; as **an attribute for the management of heterogeneous memory hierarchies**; and **as a motivation for hierarchical parallelism in compilers, over-decomposition in OS and resource management**, and for **modeling and simulation**.

- **Performance Heterogeneity:**
  - Fault resilience will also introduce inhomogeneity in execution rates as even hardware error correction is not instantaneous, and software-based resilience will introduce even larger performance heterogeneity.
- **Compiler**:
  - In an era increasingly dominated by extreme heterogeneity, hierarchical parallelism and unpredictable application behavior, possibly as the result of system-level actions to regulate heat or energy use, or to support resilience, compilers will be more important than ever
- **OS and resource management:**
  - In Complex Resource Management: One promising approach is over-decomposition, typically achieved by partitioning the problem into basic tasks with well-defined dependencies. If a running task blocks, the scheduler can launch another task on the same CPU core to minimize the idle time, assuming that the number of tasks is sufficiently large of course. An inherent advantage of this approach is its dynamic nature, in particular its ability to rapidly adjust to the shrinking or expanding set of available node resources in response to power or resilience events.
  - In Management of Heterogeneous Memory Hierarchies: Performance (BW/Latency), persistence, power, capacity, and resilience are memory attributes that runtimes will be required to manage and optimize in the near future, but the mechanisms in the OS are lacking.
- **Modeling and Simulation:**
  - Modeling and simulation (ModSim) of existing and proposed computer architectures and systems have been long-standing pillars of the ASCR portfolio [88]. In supporting these research directions, ASCR recognized the need to “meet performance, energy-efficiency, and resilience requirements of systems and applications at all scales—from embedded to exascale—recognizing their broad impacts to the larger computational science community in a range of research areas, including those affecting national security and domain sciences.”

Resilience Status&Advances in talks and other BOGs

- Bob Colwell:
  - Combination of hardware and software is needed to detect error and ensure correctness
  - Worst case is undetected errors. [The example was about a bug in a square root circuit]
- Ewa Deelman
  - Need better fault tolerance: data replication, recomputation versus retrieval
- Bruce Jacob
  - NVMM: New OS paradigms with merged VM+FS journaled main memory: built-in checkpoint restart (node local, system level)
- BOG2: Data Management:
  - System-level resilience
    - Coordination of resilience measures with other components
  - Fidelity of data
    - Understanding fidelity, trading space/accuracy for time/energy
- BOG3: Data-analytics-workflows:
  - Fidelity of data
    - Approximation uncertainty, Scientific validity, Performance variability
- BOG8: System management, admin, job scheduling
  - PRD 8.4 - Monitoring and Logging Analysis
    - Use ML to learn failure scenarios and report predicted failures to signal jobs, and state management system
    - Machine Learning databases provided by vendors and community contributed repos
    - EH component monitoring
    - Detect failures, or degraded performance
    - Functional and performance unit testing to return components to service

Resilience Status and Recent Advances in this BOG

- **Anomaly detection** (a few papers)
- **Error characterization and modeling** of extreme scale systems (several ongoing ASCR projects, publications on the topics)
- **Failure mitigation (global restart):**
    - Capturing consistent states (checkpoint) that can be used to restart execution
        - Application level checkpointing API (FTI, SCR, VeloC)
        - System level mechanism (BLCR, DMTCP- Distributed Multi-Threaded CheckPointing)
        - Deduplication, Versioning (GVR)
        - Capturing state of accelerators (GPUs, some literature ont that)
    - Storage hierarchy for checkpoint
        - Multi-level checkpointing API (FTI, SCR, VeloC)
    - Optimal checkpoint intervals: a lot of literature on this topic for single level and multilevel checkpointing
- **Failure mitigation (local restart):**
    - Task based programming/runtime env. (OMPSs)
    - Fault tolerance protocol for local restart (message logging)
    - Algorithm Based Fault Tolerance (algorithmic level)
    - ULFM (augmented MPI allowing partial restart): used directly by application or through framework (FENIX)
- **Failure mitigation (Replication)**
    - MPI implementations for Replication (RMPI, RedMPI), Literature on partial replication
- **SDC detection:**
    - Replication (Red-MPI)
    - On-line Data Monitoring Algorithms (several papers on that)
    - ABFT algorithms
    - Formal modeling/bounding of impact of errors/approximations for various program classes
    - Detectors for specific application components (loop invariants, memory accesses, address generation, etc.)
- **Error/failure containment/mitigation at programming level**
    - Containment domains
    - Runtime-based construction of consistent checkpoints for specific programming systems
    - Localizing the impact of a fault and load-balanced recovery for task-based programming/runtime models (few publications in scientific computing domain)
    - Resilience design patterns (early research)
- **Exascale:** Combination of ECP hardware and ECP software capabilities addresses resilience issues of projected Exascale Systems

Resilience: Challenges&Opportunities Specific to EHS

Heterogeneity within nodes (accelerators, memory) but also at node level (specific deep learning nodes) and potentially systems level (quantum/neuromorphic systems)

- Hardware: **Evolution towards extreme heterogeneity**
  - Increasing diversity of memory (DRAM, NVMe, etc.) and accelerators that have different reliability and error/failure modes.
  - Traditional notions of correct computation, program state, determinism and reproducibly may no longer hold true
  - Dynamic power management impacts resource reliability. For example, studies show DVFS could lead to 3 times higher failure rate
  - Applications and users may use specialized formats (e.g. MSFP9) and approximate computing for performance and to save power
- **System software: Failure mitigation at system/runtime level**
  - Capturing consistent states (checkpoint) that can be used to restart execution
    - Capturing state of accelerators (ASICs, FPGAs, in memory processing, FPGA in network, Neuromorphic)?
    - Capturing consistent states between accelerators (checkpointing protocols: how to coordinate the capture of the state of multiple devices?)
    - Capturing/restoring state in situation of power/energy management (some accelerators are in low power mode)
    - Exploit persistent state in non-volatile memory to reconstruct a consistent state
  - Memory is increasing but not the storage bandwidth/space
  - Complex storage hierarchy for checkpoint in EHS systems (built-in checkpointing with merged VM and FS?)
  - Storage abstraction moving (at least for some users) to objects (Object stores)
- **Programming models and applications**
  - Heterogeneous applications (multi-physics, coupled code, in-situ analytics). Classic Parallel programming models (MPI, OpenMP) + tasks based programming models (accelerators) + workflow environments leads to much more complex executions and more complex programming of resilience
- **Hardware/software coordination**
  - More dedicated components (accelerators, memory) with different error/failure modes and local mitigation mechanisms + power management + OS/application level error/failure management capabilities will lead to make resilience optimization extremely complex

Resilience Capability Targets and Research Directions

- **What Resilience capabilities will be needed in the 2030+ timeframe to make productive use of increasingly heterogeneous systems for DOE mission?**
- **What research is required to get from where the capabilities are now to where to where they need to be by 2030?**

Try to keep these at a high level (think topics in a call).

Resilience Capability Targets&Research Directions(1/x)

- **Error characterization/modeling specific to Extreme Heterogeneity (including impact of power management)**
  - Modes of resilient and approximate execution and associated trade-offs
- **Fundamental understanding of the impact of various error models (random, stuck-at, etc.) on application results**
  - Understand the potential for approximation in future applications
  - Natural degree of error masking and tolerance
  - Understand constraints imposed on resilience solutions
- **Protection domains** (to contain errors and failures)
  - Error propagation and failure cascade boundaries for highly heterogeneous systems are quite complex. Protection domains may span different hardware and software components
- **Trust in results from heterogeneous systems** (to address non and systematic undetected errors)
  - Combination of hardware and software detectors to detect systematic (e.g. bugs) and non-systematic errors (e.g. SDC).
- **State reduction and optimized mapping to leverage the hierarchy and address the limitations of heterogeneous storage systems**
  - Lossy Checkpointing and the impact of it on execution results
  - Reexecution/reconstruction versus state recovery: raises issue about determinism and reproducibility
  - Adapting the application checkpoint access pattern to the heterogeneity of the storage hierarchy
- **Resilient software stack**
  - Naturally resilient algorithms (ABFT, approximate, randomized, etc.)
  - Error-model-aware compilers
  - Algorithms underpinning resilient runtime systems
  - Formal analysis approaches to aid vulnerability analysis, error propagation, etc.

Resilience Capability Targets&Research Directions(2/x)

- **New Programming Models** (to ease resilience programming)
  - High-level programming models/languages need to have resilience capabilities by design as abstract concepts.
  - Low-level programming interfaces need to expose resilience features.
- **Hardware/software coordination** (to optimize resilience performance and reduce power)
  - Monitoring tools, runtime environments and interface are needed for coordinating resilience across hardware and software components (communicating resilience capabilities and quality of service requirements, and configuring the capabilities to meet requirements (a la CIFTS fault tolerance backplane). Notion connected to Resilience portability.
- **Machine/Deep learning** (as optimization techniques for resilience)
  - For error detection, and resilience performance optimization
  - Resilience for scalable Deep Learning
- **New applications** (raising new problems specific to EH)
  - Data intensive applications, stream/real-time processing (instrument connected to simulation), Dataflow/Workflow

Summary of Resilience Possible PRD 1/2

- **Ensure the validity of scientific simulation and Data analytics in presence of errors in EH.**
  - *Respond to the increasing risk of application results alteration due to extreme heterogeneous components, error modes and data formats*
  - Heterogeneous precision (including specific precision), approximate computing vs. exact computing. Effects of variable precision arithmetic (exact computations) for application-level correctness on power and possibility to have knobs to switch between exact and approximate computing.
  - Connected to: Fidelity in BOGs 2 and 3
  - Need: Combination of hardware and software (Bob Colwell) for error detection
  - Need: Error characterization and detection
  - Need: Fundamental understanding of the impact of various error models
  - Need: Trust in results from heterogeneous systems (detection of systematic and non systematic SDCs)
  - Need: Abstractions for expressing application correctness
- **Provide portable and optimized Resilience for Workflows and Applications on EH arch.:**
  - *Respond to the increasing difficulty to program and run applications to completion efficiently on extreme scale and heterogeneous systems*
  - Need better fault tolerance for workflow: data replication, recomputation versus retrieval (Ewa Deelman)
  - Need: State reduction and optimized mapping to leverage the hierarchy and address the limitations of heterogeneous storage systems
  - Need: Programming Models and libraries with resilience capabilities by design as abstract concepts.
  - Potentially leverage new OS paradigms with merged VM+FS journaled main memory (Bruce Jacob)

Summary of Resilience Possible PRD 2/2

- **Automatic/autonomic global events monitoring, mining, analytics and mitigation to maximize EH system availability and integrity/security**
  - *Respond to the increasing complexity and challenges of maintaining extreme scale and heterogeneous system available.*
  - Need: Hardware-software interactions for event notifications and mitigation
  - Need: Coordination of resilience measures (data management) with other components (BOG2)
  - Leverage: Machine Learning/deep learning to mine events
  - Need: Low-level programming interfaces need to expose resilience features.
  - Connected to: PRD 8.4 - Monitoring and Logging Analysis of BOG8
  - Connected to security/integrity issues.
- **Resilient software stack for EH architectures**
  - *Respond to improve drastically the self robustness of applications, libraries and system software, which get worse in the context of extreme heterogeneity*
  - Need: Naturally resilient algorithms (ABFT, approximate, randomized, etc.)
  - Need: Error-model-aware compilers for multiple and new error characteristics on EH systems
  - Need: Resilience algorithms underpinning novel runtime systems needed for EH systems
  - Need: Formal analysis approaches to aid vulnerability analysis, error propagation, etc. across software modules running on multiple accelerators
  - Need: Local&Global coordination of resilience measures for EH components for efficient responses&containment
  - Protection domains to contain errors and failures (local instead of global)
  - Need: Ability to express correctness constraints at all levels of the software stack on EH systems

Summary of Resilience Possible PRD

- **Ensure the validity of scientific simulation and Data analytics in presence of errors in EH.**
  - *Respond to the increasing risk of application results alteration due to extreme heterogeneous components, error modes and data formats (note: different from UQ, V&V)*
  - Need: Trust in results from heterogeneous systems (detection of systematic and non systematic SDCs)
  - Need: Combination of hardware and software (Bob Colwell) for error detection
  - Need: Error characterization, detection and modeling
  - Need: Fundamental understanding of the impact of various error models
  - Need: Abstractions for expressing application correctness
- **Provide portable and optimized Resilience for Workflows and Applications on EH arch.:**
  - *Respond the extreme difficulty to program and run applications to completion efficiently on EH systems,*
  - Need: Improve drastically the self robustness/resilience of workflow, applications, libraries, runtime and system software leveraging formal analysis of vulnerability, the ability to express correctness constraints, naturally resilience algorithms, ABFT, programming models with resilience capabilities by design, error-model-aware compiler and protection domains to contain error and failures
  - Need: State consistent capture, reduction and optimized mapping on memory and storage hierarchies
  - Need: Local&Global coordination of resilience measures for EH components for efficient responses&containment

Resilience Example: Short title of possible research direction

- One paragraph description (3 sentence/bullet)
- Research challenges
  - Metrics for progress
- Potential research approaches and research directions
- How and when will success impact technology?